

Uma matriz unidimensional em Java é uma estrutura de dados que permite armazenar múltiplos valores de um mesmo tipo, acessados por índices numéricos. A matriz deve ser inicializada antes de ser utilizada, e o acesso aos elementos ocorre via índices começando em 0. Sobre matrizes unidimensionais em Java, assinale a alternativa correta.

- A- O tamanho de uma matriz pode ser alterado dinamicamente após sua criação.
- B- O primeiro elemento de uma matriz sempre tem índice 1.
- C- Todos os elementos de uma matriz devem ser do mesmo tipo de dados.
- D- Matrizes podem armazenar valores de diferentes tipos em uma única instância.
- E- O acesso aos elementos de uma matriz ocorre de forma aleatória, sem necessidade de índice.

Os construtores são usados para inicializar objetos e podem ser sobrecarregados para aceitar diferentes parâmetros. Qual das alternativas descreve corretamente o uso de construtores em Java?

- A- Todo construtor deve ter um tipo de retorno explícito.
- B- Uma classe pode ter apenas um único construtor.
- C- Construtores podem ser sobrecarregados para aceitar diferentes argumentos.
- D- Construtores não podem inicializar atributos diretamente.
- E- Construtores precisam ser chamados manualmente após a criação do objeto.

Os métodos `get()` e `set()` são amplamente utilizados em Java para acessar e modificar os valores dos atributos de uma classe. Eles fazem parte do princípio de encapsulamento, permitindo o controle do acesso a atributos privados. A correta implementação desses métodos garante que objetos mantenham consistência interna e obedeçam às restrições de negócio definidas na lógica da aplicação. A respeito do uso dos métodos `get()` e `set()` em classes Java, assinale a alternativa correta:

- A- O uso de métodos `get()` e `set()` dispensa a necessidade de declarar os atributos como `private`.
- B- Os métodos `set()` devem ser evitados, pois tornam todos os atributos mutáveis e comprometem o encapsulamento.

C- O método `get()` é responsável por modificar o valor de um atributo, enquanto `set()` é utilizado apenas para leitura.

D- A definição de métodos `get()` e `set()` permite o acesso controlado aos atributos privados de uma classe, preservando o encapsulamento.

E- É obrigatório que toda classe Java tenha métodos `get()` e `set()` para todos os seus atributos, independentemente da lógica do sistema.

Diversos comandos e métodos fazem parte da construção e manipulação de objetos em Java. Comandos como `this`, `new`, `super()`, `get()` e `set()` são frequentemente usados em construtores, herança, encapsulamento e criação de objetos. O uso correto de cada um desses elementos é essencial para garantir legibilidade, reutilização e funcionamento adequado do código. Sobre os comandos `this`, `new`, `super()`, `get()` e `set()` em Java, analise as afirmativas abaixo:

A- O comando `this` é utilizado para referenciar a instância atual da classe e pode ser usado para acessar atributos ou métodos dessa instância.

B- O operador `new` é usado para importar pacotes externos e acessar classes de bibliotecas padrão.

C- A chamada `super()` é utilizada para sobrescrever métodos herdados da superclasse.

D- Os métodos `get()` e `set()` são gerados automaticamente pelo compilador e não precisam ser declarados pelo programador.

E- A palavra-chave `this` deve ser evitada em construtores, pois impede o uso de parâmetros com o mesmo nome dos atributos.

O método `main()` é o ponto de entrada de uma aplicação Java. Sua assinatura deve seguir um padrão específico para que a JVM possa reconhecê-lo e executá-lo corretamente. Além disso, o método `main()` pode ser sobrecarregado, mas apenas a versão com a assinatura esperada pela JVM será executada ao iniciar o programa. Modificadores, parâmetros e tipos de retorno também desempenham papéis importantes em sua definição. Sobre o método `main()` em Java, assinale a alternativa correta:

A- O método `main()` pode ser definido com qualquer nome, desde que esteja presente em uma classe pública.

B- O método main() não pode ser declarado como final, pois isso impede a execução da aplicação.

C- O método main() pode ser sobrecarregado com diferentes parâmetros, e a JVM escolherá automaticamente a versão mais adequada para executar.

D- O método main() precisa obrigatoriamente ser public e static, com retorno void e aceitar um array de String como parâmetro.

E- O método main() pode ser omitido em programas que não realizam entrada e saída de dados, desde que a classe possua um construtor público.

O laço for-each é uma forma simplificada e segura de percorrer elementos de um array em Java. Embora facilite a leitura do código e evite erros comuns relacionados à manipulação de índices, seu uso apresenta restrições importantes quando se deseja alterar o conteúdo do array ou obter a posição dos elementos durante a iteração.

A respeito do uso do laço for-each com arrays em Java, assinale a alternativa correta:

A- O laço for-each permite acessar e modificar diretamente os índices de um array durante a iteração.

B- O for-each substitui a estrutura do array por uma lista encadeada temporária, permitindo maior flexibilidade.

C- O uso de for-each é limitado a arrays unidimensionais e não pode ser aplicado a arrays multidimensionais.

D- O laço for-each permite acessar os elementos do array sequencialmente, mas não permite alterar diretamente os valores armazenados no array.

E- O uso de for-each para percorrer arrays pode causar exceções em tempo de execução, mesmo quando o array é válido e não nulo.

Em aplicações Java bem estruturadas, o uso de pacotes (packages) é fundamental para a organização e modularização do código. Pacotes servem para agrupar classes relacionadas, evitar conflitos de nomes e facilitar o controle de acesso. Para declarar corretamente um pacote, é necessário respeitar a estrutura de diretórios e a sintaxe da declaração de pacote no topo do arquivo .java.

Considere que um projeto está organizado com a seguinte estrutura de diretórios:

/projeto/

```
|— src/
|  |— br/
|  |  └─ universidade/
|  |  └─ sistema/
|  |      └─ dominio/
|  |          └─ Aluno.java
|  |          └─ controle/
|  |          └─ CadastroAluno.java
```

A classe CadastroAluno depende da classe Aluno e precisa acessá-la corretamente.

Considerando a estrutura apresentada, assinale a alternativa que representa corretamente as declarações de pacote e importação nos arquivos Aluno.java e CadastroAluno.java.

A- Em Aluno.java: package dominio; // Em CadastroAluno.java: package controle; import dominio.Aluno;

B- Em Aluno.java: package br.universidade.sistema.controle; // Em CadastroAluno.java: package br.universidade.sistema.dominio;

C- Em Aluno.java: package br.universidade.sistema.dominio; // Em CadastroAluno.java: package br.universidade.sistema.controle; import br.universidade.sistema.dominio.Aluno;

D- Em Aluno.java: package sistema.dominio; // Em CadastroAluno.java: package sistema.controle; import sistema.dominio.Aluno;

E- Em Aluno.java: package br.universidade.sistema; // Em CadastroAluno.java: package br.universidade.sistema; import br.universidade.sistema.Aluno;

Um objeto em Java é uma instância de uma classe e precisa ser criado explicitamente usando o operador new. Sobre a instanciação de objetos em Java, assinale a alternativa correta.

- A- Objetos podem ser utilizados antes de serem instanciados.
- B- O operador new aloca memória e retorna uma referência ao objeto criado.
- C- Uma variável de referência a um objeto contém diretamente o objeto em si.
- D- A JVM cria automaticamente instâncias de todas as classes na inicialização do programa.
- E- Em Java, um objeto pode ser criado sem o uso do operador new.

A classe Scanner, presente no pacote java.util, é amplamente utilizada para realizar leitura de entrada de dados em Java, seja a partir do teclado, arquivos ou outras fontes de dados. Apesar de sua simplicidade, o uso do Scanner exige atenção a detalhes como o tipo de dado lido, a quebra de linha residual e o encerramento do fluxo, que pode afetar outras leituras no programa. A respeito do funcionamento e uso da classe Scanner em Java, assinale a alternativa correta:

- A- A classe Scanner pode ser utilizada para ler entradas do teclado, mas não pode ser usada para leitura de símbolos especiais.
- B- O uso do método nextLine() após nextInt() ou nextDouble() pode gerar comportamentos inesperados devido à manipulação da quebra de linha.
- C- O método close() da classe Scanner não deve ser chamado em nenhum caso, pois causa vazamento de memória.
- D- O método NextLine() verifica se a próxima linha de entrada contém caracteres numéricos.
- E- Uma instância de Scanner criada para leitura do teclado pode ser compartilhada entre vários métodos sem qualquer restrição de uso.

A instrução break é utilizada para interromper a execução de um loop antes que a condição termine, enquanto a instrução continue ignora a iteração atual e passa para a próxima. Sobre as instruções break e continue em Java, assinale a alternativa correta.

A- break interrompe um loop completamente, enquanto continue pula a próxima execução do programa.

B- break encerra o loop imediatamente, enquanto continue faz com que a próxima iteração do loop seja executada.

C- continue interrompe um loop completamente, enquanto break apenas pula uma iteração.

D- break e continue são equivalentes e podem ser usados de forma intercambiável.

E- break só pode ser usado dentro de loops for, enquanto continue pode ser usado em qualquer estrutura de repetição.

Em Java, as estruturas de condição (if-else, switch) e de repetição (for, while, do-while) são fundamentais para o controle do fluxo de execução dos programas. Cada uma possui características e regras específicas de avaliação lógica e sintática. O uso correto dessas estruturas depende de uma compreensão precisa sobre momentos de avaliação de condição, controle de execução e adequação ao problema. Sobre as estruturas de condição e repetição na linguagem Java, assinale a alternativa correta:

A- A estrutura while garante que o bloco de código será executado pelo menos uma vez, mesmo que a condição inicial seja falsa.

B- A estrutura do-while é preferida em situações onde a verificação da condição deve ser feita antes da primeira execução.

C- A estrutura switch pode ser usada apenas com tipos primitivos int, byte, short e char, não sendo compatível com String.

D- As estruturas if-else e switch são equivalentes em todos os casos, podendo ser sempre substituídas entre si sem impacto no comportamento.

E- A estrutura for tradicional é mais adequada que for-each quando há necessidade de modificar valores ou acessar posições específicas de um array.

Em Java, o tipo char é um tipo primitivo utilizado para armazenar um único caractere, enquanto String é uma classe utilizada para armazenar sequências de caracteres. Strings podem ser concatenadas utilizando o operador +, e a classe String fornece diversos métodos úteis para manipulação de textos. Sobre o uso de char e String em Java, assinale a alternativa correta*

A- O tipo char pode armazenar uma palavra inteira, desde que esteja entre aspas simples.

B- Strings em Java são mutáveis, ou seja, seu valor pode ser alterado diretamente após a criação.

C- O operador + pode ser usado para concatenar Strings, mas não pode ser utilizado para unir char.

D- Strings são objetos, enquanto char é um tipo primitivo.

E- A classe String não possui métodos para manipulação de texto, exigindo o uso de char para esse propósito.

Quando um objeto é instanciado, seus atributos podem receber valores padrão, caso não sejam inicializados explicitamente. Considere a classe abaixo:

```
public class Exemplo {  
  
    int numero;  
  
    double valor;  
  
    String texto;  
  
}
```

Se um objeto da classe for instanciado sem atribuir valores aos atributos, quais serão os valores padrão?

A- numero = null, valor = 0.0, texto = "".

B- numero = 0, valor = 0.0, texto = null.

C- numero = 1, valor = 1.0, texto = null.

D- numero = 0, valor = null, texto = "".

E- numero = null, valor = null, texto = null.

A Programação Orientada a Objetos (POO) permite modelar sistemas complexos através de classes e objetos. Em Java, uma classe é um modelo para a criação de objetos, enquanto uma instância representa um objeto específico dessa classe. Cada instância possui atributos e comportamentos definidos pela classe. Sobre a relação entre classes e objetos em Java, assinale a alternativa correta.

A- Um objeto pode pertencer a múltiplas classes simultaneamente.

B- Uma classe define o comportamento e a estrutura dos objetos que serão criados a partir dela.

C- Em Java, objetos podem ser criados sem a necessidade de uma classe.

D- Atributos de uma classe não podem ser alterados após a criação de um objeto.

E- Métodos de uma classe não podem ser herdados por suas subclasses.

Em Java, tanto os arrays quanto a classe ArrayList são usados para armazenar coleções de elementos. Arrays são estruturas de tamanho fixo e fazem parte da linguagem desde suas primeiras versões. Já ArrayList é uma implementação da interface List da API de coleções, oferecendo operações mais flexíveis e dinâmicas para manipulação de elementos. A respeito das diferenças entre arrays e ArrayList na linguagem Java, assinale a alternativa correta:

A- Arrays e ArrayList possuem o mesmo comportamento quanto à manipulação de tamanho e remoção de elementos.

B- A estrutura ArrayList permite armazenar tipos primitivos diretamente, enquanto arrays não.

C- Arrays são mais indicados quando se deseja flexibilidade para adicionar ou remover elementos dinamicamente.

D- O uso de ArrayList requer o uso de métodos específicos para adicionar, remover e acessar elementos, ao contrário dos arrays, que utilizam indexação direta.

E- Ambas as estruturas pertencem ao pacote java.util e implementam a interface List.

Em Java, o modificador private restringe o acesso de atributos e métodos à própria classe, enquanto public permite o acesso de qualquer outra classe, independentemente do pacote. Considere que as duas classes a seguir pertencem ao mesmo pacote:

```
public class Conta {  
  
    private double saldo;  
  
    public void depositar(double valor) {  
  
        saldo += valor;  
  
    }  
  
}  
  
public class Banco {  
  
    public static void main(String[] args) {  
  
        Conta c = new Conta();  
  
        c.depositar(1000);  
  
        // System.out.println(c.saldo);  
  
    }  
  
}
```

}

Assinale a alternativa correta:

- A- O método depositar() só pode ser chamado dentro da própria classe Conta.
- B- A linha comentada está correta, pois saldo é um atributo numérico.
- C- A linha comentada não compila, pois saldo é private e não acessível fora da classe Conta.
- D- Atributos private podem ser lidos diretamente por métodos public de outras classes.
- E- A visibilidade private se aplica apenas a métodos, não a atributos.

Os loops são usados em Java para repetir um bloco de código até que uma condição seja satisfeita. Os principais tipos são:

- **for: usado quando o número de repetições é conhecido.**
- **while: usado quando o número de repetições é desconhecido.**
- **do-while: garante que o bloco de código seja executado pelo menos uma vez.**

Qual das alternativas descreve corretamente a diferença entre os loops while e do-while em Java?

- A- O loop while sempre executa pelo menos uma vez, enquanto o do-while pode não executar nenhuma vez.
- B- O loop while verifica a condição antes da execução do bloco, enquanto o do-while verifica após a primeira execução.
- C- Ambos os loops sempre executam pelo menos uma vez.
- D- O loop do-while só pode ser usado para contadores numéricos.
- E- O loop while é usado apenas quando há um número fixo de iterações.

O construtor é um método especial de uma classe que é chamado quando um objeto é criado. Qual das alternativas está correta sobre construtores em Java?

A- Todo construtor deve ter um tipo de retorno explícito.

B- Uma classe pode ter apenas um único construtor.

C- Construtores podem ser sobrecarregados para aceitar diferentes conjuntos de parâmetros.

D- Construtores são métodos comuns e podem ser chamados diretamente como qualquer outro método.

E- Construtores são opcionais e nunca são executados automaticamente.

O Java organiza suas classes em pacotes para facilitar a modularização e reutilização do código. Para acessar classes de um pacote diferente, utiliza-se a instrução import. Qual das alternativas descreve corretamente o uso da instrução import em Java?

A- O import permite o uso de classes externas sem a necessidade de especificar seu pacote completo.

B- A instrução import java.util.*; importa todas as classes do Java, incluindo pacotes personalizados.

C- O import é obrigatório para utilizar qualquer classe em Java.

D- Um import só pode ser utilizado dentro do método main.

E- A instrução import static permite importar apenas métodos privados de outras classes.

A classe ArrayList é uma estrutura de dados dinâmica em Java que permite armazenar e manipular coleções de objetos. Diferente das matrizes, o ArrayList pode ter seu tamanho ajustado dinamicamente. Sobre a classe ArrayList em Java, assinale a alternativa correta.

A- ArrayLists podem armazenar apenas valores primitivos como int e double.

B- O tamanho de um ArrayList é fixo após sua inicialização.

- C- ArrayLists podem conter apenas objetos, não tipos primitivos.
- D- O método size() retorna o número de elementos vazios dentro da lista.
- E- Para usar um ArrayList, não é necessário importar nenhuma classe.

ignore as alternativas que estão com a cor diferente, não tenho certeza do gabarito, foram apenas as questões que eu marquei na prova.